Module 1 - Lecture 6

Introduction to Objects

Review

- Command Line programs
- Reading in data
- Parsing data
- Writing data to the console

Objects!

What is an Object?

An **object** is an in-memory data structure that combines state and behavior into a useful abstraction.

Classes

A **class** is a grouping of variables and methods in a source code file from which we can generate objects.

Class

Object

Object

Creating an Object

Declare:

```
    House myHouse;
```

Instantiate and Initialize:

```
    myHouse = new House();
    myHouse = new House("Red");
```

All in one:

```
    House myHouse = new House("Red");
```

There is that **new** keyword again.

Memory Management!

Primitive vs. Reference Types

Primitive or Value Types

boolean

char

byte
int
long
float
decimal

Reference Types

Strings

Arrays

Objects

… really everything other than primitives.

Primitive vs. Reference Types

Primitive or Value Types

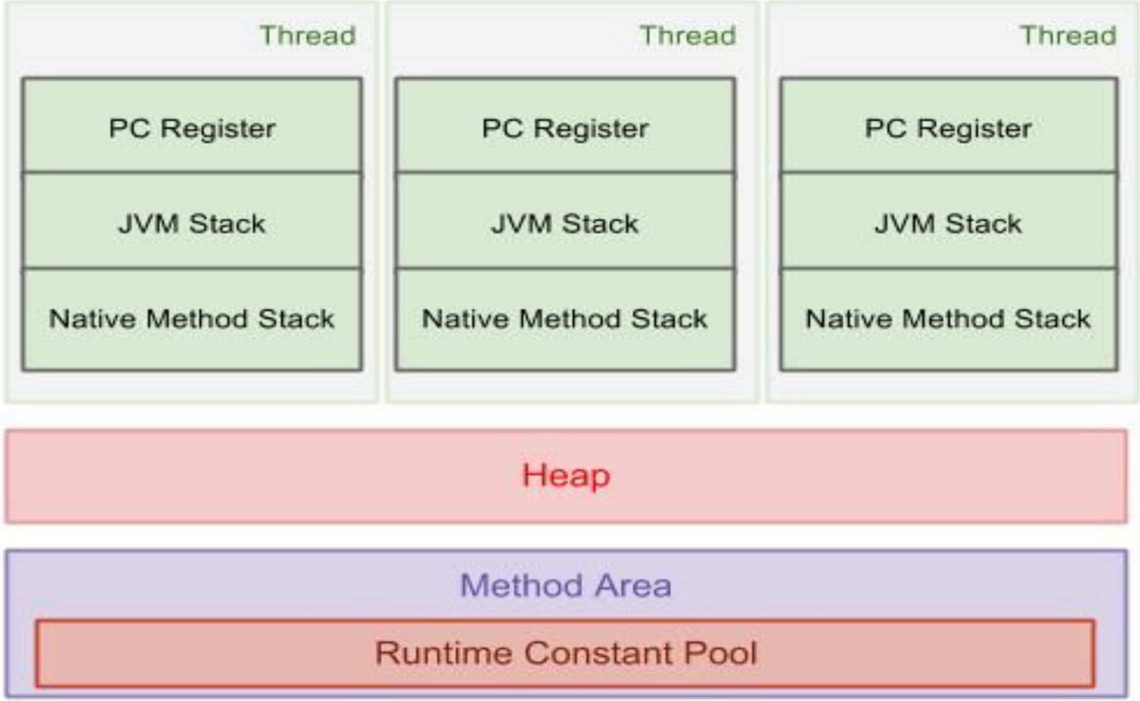
Type	Size (bits)
boolean	1
char	16
byte	8
int	32
long	64
float	32
double	64

Reference Types

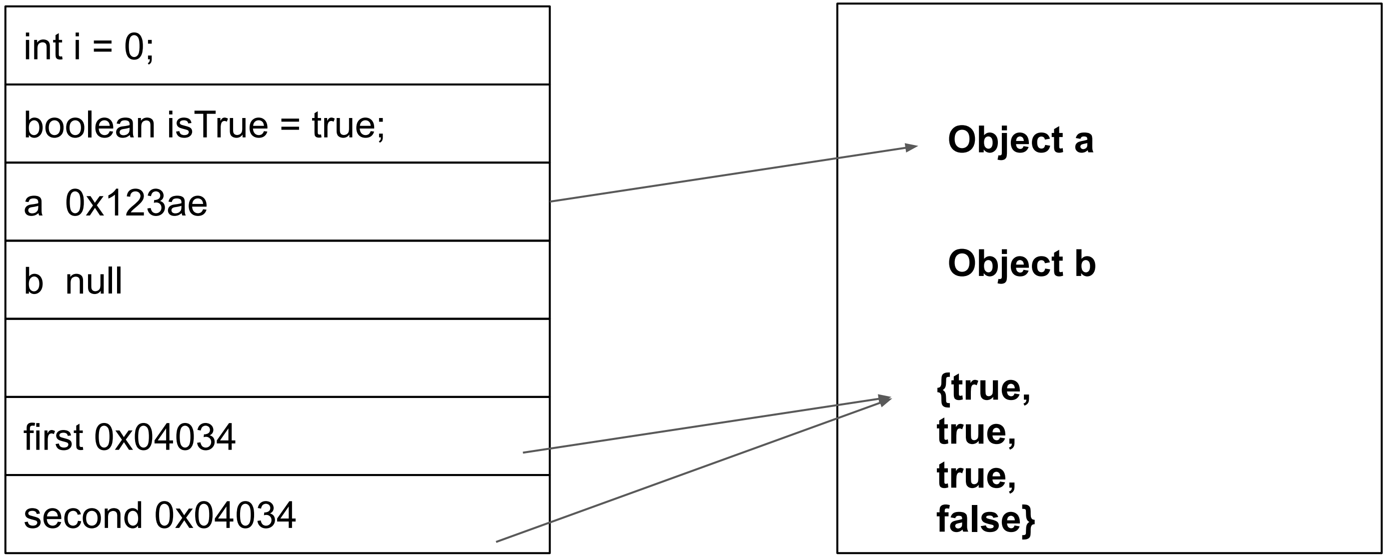
Type	Size (bits)
Strings	???
Arrays	???
Objects	???

… really everything other than primitives.

Memory Layout



Stack vs. Heap



int i = 0;

boolean isTrue = true;

a 0x123ae

b null

first 0x04034

second 0x04034

Object a

Object b

{true,
true,
true,
false}

Stack

Heap

Reference type assignment

What will be printed by the code below?

```
boolean[] first = { true, true, false, false };

boolean[] second = first;

second[2] = true;

System.out.print(first[2]);
```

Strings

- They are an object, but they are unique.
  - Do not require the **new** keyword, but can use it.
  - Immutable
  - Are pooled by Java

- Like other objects, they have methods and properties that we can access.

Let’s Code!

Reading

- Module 1
  - Collections Part 1

QUESTIONS?